## AMENDMENTS

## II. IN THE CLAIMS:

The following Listing of Claims replaces all previous versions.

Listing of Claims

1.-10. (Cancelled).

11. (Currently Amended) A method comprising:

dividing a ~~thread that is to be executed~~ dynamic sequential program into multiple epochs, wherein each epoch includes 2 or more instructions;

in a redundant multi-threading (RMT) system having leading and trailing threads, redundantly executing in parallel first and second instances for each epoch ~~by separately executing corresponding leading and trailing epoch instances~~ as the leading and trailing threads in the RMT system;

for the executed first and second epoch instances, saving ~~committed results from the executed corresponding leading and trailing~~ epoch instances store results as speculative stores to memory, the speculative stores being exposed;

comparing the ~~committed results that should correspond to one another from the leading and trailing epoch instances~~ exposed stores; and

if they match, committing a single set of ~~results~~ the exposed stores ~~for the executed epoch based on the compared results~~.

12. (Currently Amended) The method of claim 11, wherein the speculative stores are from a re-order buffer ~~saved results are saved as speculative~~.

13. (Previously Presented) The method of claim 12, wherein the two or more instructions executed in the execution of the epoch instances are buffered prior to epoch execution completion.

14. (Cancelled).

15. (New) The method of claim 11, wherein the memory is L1 cache memory.

16. (New) A method, comprising:

breaking a program to be executed into multiple epochs each having two or more instructions;

redundantly executing the program by redundantly executing each epoch separately, and

sending speculative results for each epoch to memory;

checking the speculative results for each epoch against each other; and

if they match, committing the results.

17. (New) The method of claim 16, in which the speculative results are speculative stores.

18. (New) The method of claim 16, in which the memory is L1 cache memory.